SUBSTITUTE SPECIFICATION UNDER 37 C.F.R. 1.125

METHOD AND APPARATUS FOR INCREASING THE DEVICE COUNT ON A SINGLE ATA BUS

CROSS-REFERENCE TO RELATED APPLICATIONS

This application claims priority from U.S. Provisional Patent Application Serial No. 60/210,713, filed June 9, 2000 entitled "Increasing the Disk Drive Count on a Single ATA Bus" which is incorporated herein by reference in its entirety.

FIELD OF THE INVENTION

The present invention relates to increasing the device count on a single ATA bus. In particular, the present invention allows more than two devices to be connected to a single ATA bus.

BACKGROUND OF THE INVENTION

Various host computers (and most personal computers) include an Advanced Technology Attachment (ATA) controller for communicating with ATA devices using an ATA bus. The controller typically has two channels, each of which is connected to an ATA bus, and each ATA bus can be connected to two devices. The host computer can include additional controllers to connect to more devices. However, additional controllers add cost, do not allow for command overlapping between devices and do not improve performance. Furthermore, appropriate cabling is difficult.

Alternative bus protocols are available. For instance, the small computer system

interface (SCSI) allows 255 SCSI devices to be connected to a single SCSI bus. However, SCSI devices are expensive, and the SCSI controller is not normally provided in a personal computer. Furthermore, although SCSI improves performance over ATA, the SCSI data rate cannot be fully exploited by most host computers and SCSI devices.

Accordingly, it would be advantageous to increase the number of devices that can be connected to a single ATA bus.

SUMMARY OF THE INVENTION

The present invention allows more than two devices to be connected to a single ATA bus.

In accordance with an embodiment of the present invention, the devices are each assigned a unique identifier, and the controller selects a device by sending a selection command that includes a selection identifier across data lines of the ATA bus to the devices. The devices each receive the selection command and compare the selection identifier to the assigned identifier. The device which matches the selection identifier to its assigned identifier is selected, and the other devices are not selected. The selected device prepares to receive an additional command or data from the ATA bus, and the other devices disconnect themselves from the ATA bus.

In accordance with another embodiment of the present invention, the selected device updates its status register and asserts PDIAG and INTRQ in response to the selection command, and the controller reads the status register of the selected device to verify proper device selection in response to the PDIAG and INTRQ assertions.

In accordance with another embodiment of the present invention, the controller

includes DASP, PDIAG and INTRQ registers, the ATA bus includes multiple DASP, PDIAG and INTRQ lines connected to the DASP, PDIAG and INTRQ registers, and the devices share the data lines but have dedicated DASP, PDIAG and INTRQ lines.

In accordance with another embodiment of the present invention, the controller includes DASP, PDIAG and INTRQ registers, the ATA bus includes multiple DASP, PDIAG and INTRQ lines connected to the DASP, PDIAG and INTRQ registers, and the devices share the data lines and the DASP, PDIAG and INTRQ lines.

Advantageously, the present invention allows more than two devices to be connected to a single ATA bus with only slight modifications to the bus system.

Additional advantages of the present invention will become readily apparent from the following discussion, particularly when taken together with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a block diagram illustrating a computer system that includes a bus system with a controller, an ATA bus and devices;

Fig. 2 is a flowchart illustrating operational steps taken during power-up of the bus system without additional hardware;

Fig. 3 is a flowchart illustrating operational steps taken by a device in response to a selection command that includes determining whether the bus system includes additional hardware;

Fig. 4 is a flowchart illustrating operational steps taken during device selection without additional hardware in the bus system;

Fig. 5 is a block diagram illustrating the additional hardware in the bus system;

Fig. 6 is a flowchart illustrating operational steps taken during power-up of the bus system with the additional hardware; and

Fig. 7 is a flowchart illustrating operational steps taken during device selection with the additional hardware in the bus system.

DETAILED DESCRIPTION

Fig. 1 illustrates in block diagram form a computer system **100** that includes a host computer **104** and a bus system **106**. The host computer **104** includes a system bus **108**, a processor **112**, a system memory **116** and a controller **120**. The bus system **106** includes the controller **120**, a single ATA bus **124** and devices **128** designated as devices **128-0**, **128-1**, **128-2**, **128-3**, **128-4**, **128-5**, **128-6** and **128-7**. Thus, the host computer **104** and the bus system **106** share the controller **120**.

The controller **120** provides an ATA interface that connects the system bus **108** to the ATA bus **124**. Thus, the controller **120** translates between the protocols of the system bus **108** and the ATA bus **124**. The controller **120** also connects the system bus **108** to the devices **128** via the ATA bus **124**.

The ATA bus **124** includes the ATA signal lines such as data lines and the DASP (device active, slave present), PDIAG (passed diagnostic) and INTRQ (interrupt request) lines under the ATA protocol. Likewise, the signal lines are terminated under the ATA protocol. The ATA bus **124** also includes a ribbon cable, a ribbon connector connected to the controller **120** and edge connectors connected to the devices **128**. However, unlike a conventional ATA bus, the ATA bus **124** has eight edge connectors (rather than two) to

provide separate edge connectors for each of the eight devices **128**.

The devices **128** are ATA storage devices such as hard disk drives, floppy drives, optical drives and tape drives. The devices **128** include ATA connection pins such as data, DASP, PDIAG and INTRQ connection pins for connecting with the corresponding signal lines in the ATA bus **124**. Furthermore, the devices **128** are each assigned a unique identifier. The identifiers may be assigned through hardware or software. For example, the identifiers may be assigned using jumpers on the exteriors of the devices **128**. Alternatively, the identifiers may be assigned by setting software switches during the initialization of the devices **128**.

The host computer **104** can perform ATA command and data transfer operations across the ATA bus **124** with any of the devices **128**. The controller **120** sends a selection command that includes a selection identifier over the data lines to the devices **128**. The selection command can be issued at any time the data lines are available, and need not be issued during a selection phase, a command phase or a resolution phase. The devices **128** share the data lines and therefore receive the selection command and its selection identifier from the ATA bus **124**. The devices **128** each compare the selection identifier to the assigned identifier. For example, the devices **128** include a processor with firmware that compares the selection identifier to the assigned identifier. If a device **128** is selected (the selection identifier matches the assigned identifier), then it sends a confirmation signal over the ATA bus **124** to the controller **120**. The remaining devices **128** are not selected (the selection identifier does not match the assigned identifier) and disconnect from the ATA bus **124**. The controller **120** then exchanges information (such as commands and data) over the ATA bus **124** with the selected device **128**. The

controller **120** can subsequently de-select the selected device **128** and select another device **128** by issuing another selection command with a different selection identifier over the data lines.

In an embodiment, the controller **120** is a conventional ATA controller with modified firmware, the ATA bus **124** is a conventional ATA bus with eight edge connectors, and the devices **128** are conventional ATA devices with modified firmware. In another embodiment, the controller **120** is a conventional ATA controller with modified firmware and additional hardware (such as control registers and logic gates), the ATA bus **124** is a conventional ATA bus with eight edge connectors and additional hardware (such as separate DASP, PDIAG and INTRQ lines for each edge connector), and the devices **128** are conventional ATA devices with modified firmware. In yet another embodiment, the devices **128** determine whether the additional hardware is present and respond to the selection command based on the determination.

In an embodiment, the controller **120** selects the devices **128** by asserting a single selected data line, and the ATA bus **124** includes N data lines and N edge connectors. In this instance, the ATA bus **124** can support as many as N devices **128**. In another embodiment, the controller **120** selects the devices **128** by sending a binary number across the data lines, and ATA bus **124** includes N data lines, N DASP lines, N PDIAG lines, N INTRQ lines, 2^N edge connectors and decoding logic that converts the DASP, PDIAG and INTRQ signals at the edge connectors into a binary number that is sent across the data lines to the controller **120**. In this instance, the ATA bus **124** can support as many as 2^N devices **128**.

Fig. 2 illustrates power-up of the bus system **106**. Initially, the power is turned on

(step **200**). The devices **128** that are not logical unit zero (not designated as the master) tristate the data lines, INTRQ and PDIAG (step **204**). To tristate a signal line means that the device **128** sets its driver for that signal line to a high impedance condition, as is well-known under the ATA protocol. The device **128** that is the logical unit zero then tristates PDIAG (step **208**).

A determination is made as to whether PDIAG has been negated (the devices **128** have all tristated PDIAG) (step **212**). If not, the bus system **106** waits. If so, the device **128** that is logical unit zero asserts PDIAG, clears its BSY (busy) bit in its status register and sets its DRDY (device ready) bit in its status register (step **216**). A determination is then made as to whether the BSY bit is cleared and the DRDY bit is set (step **220**). If not, the bus system **106** waits. If so, power-up of the bus system **106** is complete (step **224**).

Fig. 3 is a flowchart illustrating the response of a device **128** to a selection command. Initially, a selection command that includes a selection identifier is sent on data lines of the ATA bus **124** (step **300**). Each device **128** determines whether the selection identifier matches its assigned identifier (step **304**).

If a device **128** determines that the selection identifier matches its assigned identifier, the selected device **128** next determines whether the bus system **106** includes additional hardware (step **308**). In particular, the selected device **128** determines whether the controller **120** includes additional control registers. If so, the selected device **128** enables its data lines drivers (step **312**) and asserts DASP (step **316**). If not, the selected device **128** tristates PDIAG (step **320**), determines whether PDIAG is negated and idles until PDIAG is negated (step **324**). Once PDIAG is negated, the selected device **128** asserts PDIAG (step **328**), enables its data lines and INTRQ drivers (step **332**) and asserts

INTRQ (step **336**).

If a device **128** determines that the selection identifier does not match its assigned identifier (step **304**), the device **128** determines whether the bus system **106** includes additional hardware (step **340**). If so, the device **128** tristates the data lines (step **344**) and negates DASP (step **348**). If not, the device **128** tristates the data lines, INTRQ and PDIAG (step **352**).

Fig. 4 is a flowchart illustrating the selection of a device **128** in the bus system **106** without the additional hardware. Initially, the host computer **104** issues a selection command that includes a selection identifier through the controller **120** to select one of the devices **128** (device n) (step **400**). The devices **128** receive the selection command over the data lines of the ATA bus **124** (step **404**). The devices **128** whose assigned identifiers do not match the selection identifier are not selected and tristate the data lines and INTRQ and negate PDIAG (step **408**).

The selected device **128** whose assigned identifier matches the selection identifier tristates PDIAG (step **412**), determines whether PDIAG is negated and idles until PDIAG is negated (step **416**). Once PDIAG is negated, the selected device **128** asserts PDIAG and enables its data lines and INTRQ drivers (step **420**) and asserts INTRQ (step **424**).

The controller **120** determines whether INTRQ is asserted (step **428**). If not, the controller **120** waits. If so, the controller **120** reads the status register in the selected device **128** to verify the successful selection of the selected device **128** (step **432**). The host computer **104** then issues an additional command, such as to send or receive data, through the controller **120** to the selected device **128** (step **436**). Following a read/write command to send or receive data, data may be transferred between the controller **120** and

the selected device **128**.

Fig. 5 shows the additional hardware in the bus system **106**.

The controller **120** includes the control registers **500**, an AND gate **504**, an OR gate **508**, an interrupt request signal line **512**, an interrupt mask register bus **544** and an interrupt logic bus **548**. The control registers **500** include a selected status register **516**, a ready status register **520**, an interrupt pending register **524** and an interrupt mask register **528**. The control registers **500** may be accessed through the register access address (control block register 4) and the register access data (control block register 5) under the ATA protocol. The AND gate **504** is eight two-input AND gates, the OR gate **508** has eight inputs and a single output, the interrupt request signal line **512** is a single signal line and the interrupt mask register bus **544** and the interrupt logic bus **548** are eight-bit buses.

The ATA bus **124** includes the data bus **502** that includes the data lines, the DASP bus **532** that includes eight DASP lines (DASP 0-7), the PDIAG bus **536** that includes eight PDIAG lines (PDIAG 0-7) and the INTRQ bus **540** that includes eight INTRQ lines (INTRQ 0-7). Thus, the data bus **502**, the DASP bus **532**, the PDIAG bus **536** and the INTRQ bus **540** are eight-bit buses.

The DASP lines, the PDIAG lines and the INTRQ lines are dedicated to the individual devices **128**. For example, DASP line 0, PDIAG line 0 and INTRQ line 0 are connected to device **128-0** and disconnected from devices **128-1** to **128-7**, DASP line 1, PDIAG line 1 and INTRQ line 1 are connected to device **128-1** and disconnected from devices **128-0** and **128-2** to **128-7**, and so on. Thus, unlike a conventional ATA bus which includes single DASP, PDIAG and INTRQ lines that are shared by the devices, the ATA bus **124** includes multiple DASP, PDIAG and INTRQ lines that are not shared by

the devices **128**. Instead, the devices **128** are each connected to a separate DASP, PDIAG and INTRQ line.

The DASP bus **532** is an input to the selected status register **516.** Accordingly, each device **128** selectively drives a corresponding DASP line on the DASP bus **532** to set a corresponding bit in the selected status register **516**. For example, device **128-0** selectively drives DASP line 0 to set a corresponding bit in the selected status register **516**, device **128-1** selectively drives DASP line 1 to set a corresponding bit in the selected status register **516**, and so on.

The PDIAG bus **536** is an input to the ready status register **520**. Accordingly, each device **128** selectively drives a corresponding PDIAG line on the PDIAG bus **536** to set a corresponding bit in the ready status register **520**. For example, device **128-0** selectively drives PDIAG line 0 to set a corresponding bit in the ready status register **520,** device **128-1** selectively drives PDIAG line 1 to set a corresponding bit in the ready status register **520**, and so on.

The INTRQ bus **540** is an input to the interrupt pending register **524.** Accordingly, each device **128** selectively drives a corresponding INTRQ line on the INTRQ bus **540** to set a corresponding bit in the interrupt pending register **524**. For example, device **128-0** selectively drives INTRQ line 0 to set a corresponding bit in the interrupt pending register **540,** device **128-1** selectively drives INTRQ line 1 to set a corresponding bit in the interrupt pending register **540,** and so on.

The interrupt mask register **528** is written by the controller **120**. Typically, only one bit of the interrupt mask register **528** is set and the remaining bits of the interrupt mask register **528** are cleared. The set bit indicates which one of the devices **128** is the

selected device **128**.

The INTRQ bus **540** and the interrupt mask register bus **544** are inputs to the AND gate **504**. The INTRQ bus **540** provides interrupt requests from the devices **128** to the AND gate **504**, and the interrupt mask register **528** indicates the selected device **128** to the AND gate **504**. The output of the AND gate **504** is sent across the interrupt logic bus **548** to the input of the OR gate **508**, and the output of the OR gate **508** is sent across the interrupt request signal line **512** to the system bus **108** and in turn to the processor **112**. If a set bit of the interrupt mask register **528** matches an asserted signal on the INTRQ bus **540** then a signal line on the interrupt logic bus **548** will be high, and therefore the output of the OR gate **508** will also be high and the interrupt request signal line **512** will pass the interrupt request to the processor **112**. Thus, a high signal from the OR gate **508** indicates that the selected device **128**, as confirmed by the selection stored in the interrupt mask register **528**, is generating an interrupt request.

In an alternative embodiment, the ATA bus **124** may include decode logic to set the appropriate bits in the control registers **500**, and each device **128** may be provided with a driver connected to each of the signal lines in the DASP bus **532**, the PDIAG bus **536** and the INTRQ bus **540**. That is, the DASP lines, the PDIAG lines and the INTRQ lines are shared by (rather than dedicated to) the individual devices **128**. For example, DASP line 0, PDIAG line 0 and INTRQ line 0 are connected to devices **128-0** to **128-7**, DASP line 1, PDIAG line 1 and INTRQ line 1 are connected to devices **128-0** to **128-7**, and so on. Thus, unlike a conventional ATA bus which includes single DASP, PDIAG and INTRQ lines that are shared by the devices, the ATA bus **124** includes multiple DASP, PDIAG and INTRQ lines that are shared by the devices **128**. Furthermore, the

selection command sent over the data bus **502** and the DASP, PDIAG and INTRQ signals sent over the DASP bus **532**, the PDIAG bus **536** and the INTRQ bus **540**, respectively, can be eight-bit binary words with 2^8 or 256 addresses. As a result, as many as 256 devices **128** can be connected to the ATA bus **124** provided the ATA bus **124** has 256 edge connectors and the hardware and/or signaling protocols are altered to accommodate increased signal path lengths.

The additional hardware also provides the opportunity for overlapping commands across the devices **128**. The interrupt pending register **524** allows the processor **112** to identify each device **128** that has asserted INTRQ and therefore has a pending interrupt. For instance, if the device **128-0** asserts INTRQ line 0 and the device **128-1** asserts INTRQ line 1, thereby setting the corresponding two bits in the interrupt pending register **524**, the commands can be overlapped by issuing a first command to the device **128-0** and before the first command operation is completed issuing a second command to the device **128-1**.

Fig. 6 illustrates power-up of the bus system **106** with the additional hardware using an alternative protocol. Initially, the power is turned on (step **600**). Each device **128** asserts DASP within a selected amount of time, for example from about 1-10 milliseconds (step **604**). Each device **128** asserts PDIAG and negates DASP when it is ready to receive a command (step **608**).

The controller **120** determines whether the devices **128** have negated DASP (step **612**). If not, the controller **120** waits (step **612**). If so, the controller **120** next determines whether the devices **128** have asserted PDIAG (step **616**). If not, the controller **120** may notify the processor **112** of a problem or that fewer than the expected or possible number

of devices **128** are connected, or wait until the devices **128** have asserted PDIAG. After the devices **128** have asserted PDIAG, or after the controller **120** has been instructed or has decided to continue with less than all of the devices **128** asserting PDIAG, the host computer **104** may read or write to the ATA bus **124** through the controller **120** (step **620**).

Fig. 7 is a flowchart illustrating the selection of a device **128** in the bus system **106** with the additional hardware using an alternative protocol. Initially, the host computer **104** issues a selection command to select a device **128** (device n) (step **700**). The devices **128** receive the selection command over the data lines of the data bus **502** (step **704**). The devices **128** that are not selected tristate the data lines and negate DASP (step **708**). The selected device **128** enables its data lines drivers and asserts DASP (step **712**). The controller **120** determines whether only the intended device **128** is indicated as selected in the selected status register **516** (step **716**). If so, the host computer **104** may issue another command, for example a command to read or write data, to the selected device **128** through the controller **120** and over the ATA bus **124** (step **720**).

The present invention may include a back plane for implementing the physical channel of the ATA bus **124**, particularly if the additional hardware is included. The back plane minimizes the distance that signals must travel between the controller **120** and the devices **128**. The back plane also minimizes the capacitance in the signal lines between the controller **120** and the devices **128**. The back plane may incorporate the DASP bus **532**, the PDIAG bus **536**, the INTRQ bus **540**, the decode logic associated with these buses, and the AND gate **504**, the OR gate **508**, the interrupt request signal line **512** and the interrupt logic bus **548** for interrupt management.

The present invention may also include a frame for holding the devices **128** in close proximity to one another. For example, a frame for holding eight hard disk drives (the devices **128**) having integrated connectors allows each hard disk drive to be connected to the ATA bus **124** in a modular fashion. The frame is no taller than necessary in order to accommodate the eight hard disk drives. For example, the frame may result in a hard disk drive stack height of 8 ½ inches.

The foregoing discussion of the invention has been presented for purposes of illustration and description. Further, the description is not intended to limit the invention to the form disclosed herein. Consequently, variations and modifications commensurate with the above teachings, within the skill and knowledge of the relevant art, are within the scope of the present invention. The embodiments described herein are further intended to explain the best way presently known of practicing the invention and to enable others skilled in the art to utilize the invention in such or in other embodiments or with various modifications required by their particular application or use of the invention. It is intended that the appended claims be construed to include the alternative embodiments to the extent permitted by the prior art.